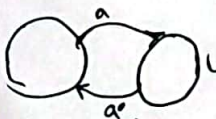


می‌دانیم $EMPTY_{DFA}$ یک زبان تصمیم‌ناپذیر است! $accept, if L(D) is infinite$
 حال، انتقاد از T که TM زبان $EMPTY_{DFA}$ است می‌شود
 حاصل می‌کنیم!

اگر K تعداد حالت‌های D برای رسیدن به حالت $accept$ در نظر بگیریم. یک DFA می‌سازیم که هر ورودی با K حرف a به D می‌فرستد. حال چون می‌دانیم DFA خاصیت بسته‌شدن عمل اشتراک را دارد، یک DFA جدید M' می‌سازیم. اگر DFA جدید \emptyset نباشد یعنی در زبان D K حرف a وجود دارد (LLD) بدون بازگشت.



state های DFA است یعنی در DFA زبان K حرف a وجود دارد هیچ حروفی وجود ندارد (روایع شکل)
 در DFA وجود دارد اگر DFA جدید \emptyset نباشد یعنی «اولی که در DFA وجود داشته است که K حرف a باشد»
 چون با تکرار حروف ما بین دو state حاضر در ورودی توانیم به زبان دلخواهی زبان را ادا کردیم پس
 اگر DFA جدید \emptyset نباشد یعنی زبان D نامتناهی است. همچنین اگر $EMPTY_{DFA}$ یک مسئله

decidable هست پس با ورودی M' DFA می‌توانیم مسئله $EMPTY_{DFA}$ را حل کنیم پس

خروجی $accept$ و $reject$ کافیست. اگر $EMPTY_{DFA}$ خروجی acc دارد، D را $reject$ می‌کنیم و در صورتی که خروجی $reject$ دارد، D را acc می‌کنیم.

آند $L(R)$ را زبان L بنامیم. باید ثابت کنیم L تصمیم نپذیراست!

حال زبانی را در نظر می‌گیریم که شامل تمام رشته‌هایی است که دارای زیررشته 111 هستند و آن را زبان

A بنامیم. $A = \{ w \in \Sigma^* \mid w \text{ has } 111 \text{ as a substring} \}$

پس $B = A \cap L$ را تعریف می‌کنیم. می‌دانیم اگر B محلی باشد یعنی L دارای هیچ زیررشته‌ای شامل 111 نمی‌شود! پس reject می‌کنیم.

همچنین می‌دانیم $EMPTY_{REG}$ یک مسئله تصمیم نپذیراست.

$D(R)$ را عنوان می‌دهیم که به ما این توانایی تصمیم‌گیری برای $EMPTY_{DFA}$ می‌دهیم. این مسئله DFA

decidable است. خودی $EMPTY$ اگر $accept$ (رد) reject می‌کند اگر $reject$ (رد) reject می‌کند.

$accept$ می‌کنیم. چون $EMPTY$ ، تصمیم نپذیراست پس R هم تصمیم نپذیراست!

(۴) می دانیم complement سوشی هر زبان TM-recognizable تنها تشخیص پذیر است!

اثبات: برای زبان TM-recognizable A ثابت می کنیم.

فرض خلف ما این است که \bar{A} نیز تشخیص پذیر باشد پس برای A و \bar{A} برای هر دو یک

تشخیص دهنده وجود دارد طبق قضیه دین اگر برای A و \bar{A} موردی باشد و تشخیص پذیرند پس non-trivial هستند

و زبان A undecidable است در طرفی می دانیم که A و \bar{A} هر دو recognizable باشند اگر A

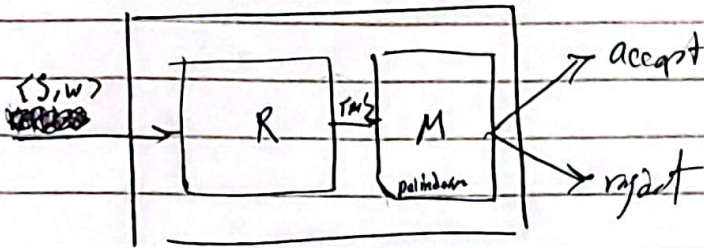
یک زبان decidable است! که این تناقض است! پس complement یک زبان recognizable

تشخیص پذیر است!

می دانیم $A \cup B$ و $A \cap B$ یعنی اشتراک یک recognizable با recognizable یا non-recognizable نیز می تواند

recognizable باشد یعنی در واقع دو مجموعه A و B recognizable تحت عمل تقاطع بسته هستند!

(5) $L(M)$ به A_{TM} کاهش می دهیم $A_{TM} \leq L(M)$



فرض کنیم M $decidable$ باشد (فرض کنید)

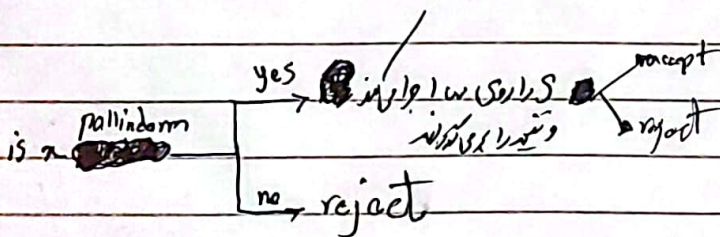
حال با کاهش A_{TM} به $L(M)$ به سادگی می بینیم که A_{TM} را نمی توانیم $decide$ کنیم که غیر ممکن است!

ورودی M' برای ساین ترنیت M می سازیم!

$$L(M) = \{ \langle M' \rangle \mid M' \text{ is a TM and } L(M') \text{ is all-palindrome} \}$$

حال باید M' را در R سازیم و به M ورودی دهیم. چون M' $decidable$ باشد و وقتی R روی $\langle S, w \rangle$ $accept$ می کند، M نیز باید $accept$ کند. پس برای $\langle S, w \rangle$ های $acceptable$ ، M' $accept$ می کند. $L(M') = \{ \langle M' \rangle \mid L(M') \text{ is all-palindrome} \}$

برای M' ، اگر ورودی M' به هم می آید:



در واقع می خواهیم M در حالتی $accept$ کند که S $accept$ کند و همچنین ورودی M' $palindrome(x)$ باشد (یا در واقع $L(M')$ مثل رشته های $palindrome$ باشد)

$$\langle S, w \rangle \in A_{TM} \Leftrightarrow S \text{ accepts } w$$

$$\Leftrightarrow L(M') = \text{palindrome}$$

$$\Leftrightarrow M \text{ accepts } M'$$

$$\Leftrightarrow M' \in L(M)$$

با فرض $decidable$ بودن M و A_{TM} $decidable$ کنیم که سادگی است

(۶) الف) decidable است

می‌دانیم TM های که حتماً tape در آنها TM های استاندارد هستند. حال نزدیک TM با ۳ تا tape استفاده می‌کنیم. یک tape ورودی (w) یک tape دیگر transition ها و در tape نایم نیز همی state های TM. صورت گذشته قرار دارند و شده بایک علامت خاص مثل # در هم جدا شده اند. هر بار که با خواندن ورودی از w به حالت جدیدی ورودی سوئیچ روی اولین حرف گذشته ای که state یک علامت خاص می‌شود (تقطعی اندازیم)

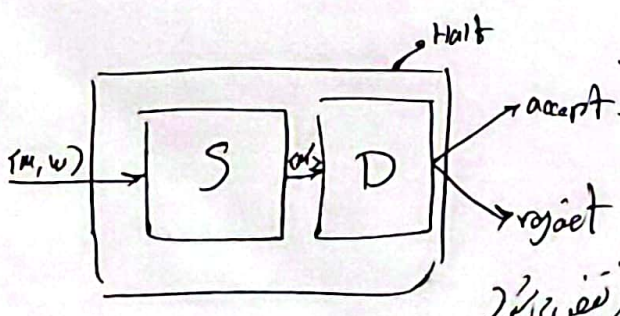
هر بار در وقتی که وارد یک state می‌شیم (تایم می‌شود خواندن کامل w) اگر قبلاً علامت خورده بود $\langle M, w \rangle$ را accept می‌کنیم و در غیر این صورت reject می‌کنیم.

امکان است با این حل نوی $1 \leq p$. بنفیس وجود ندارد چرا که در وقتی که یک state را می‌بینیم از دیدار مدافعت کنیم، اجرای TM توقف و یا منقطع می‌شود accept (ارسال می‌شود پس امکان افتادن در $1 \leq p$ وجود ندارد $\Rightarrow L$ یک زبان تصمیم ناپذیر است)

undecidable است (HALT $\leq D$)

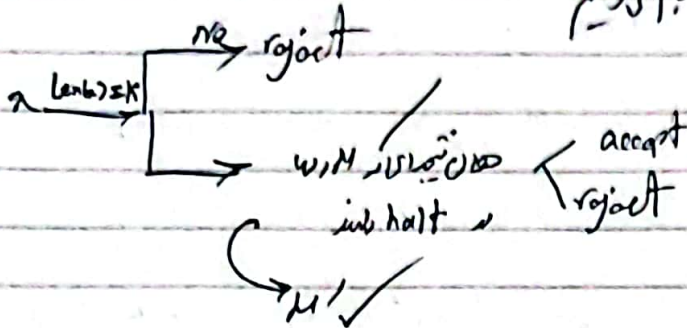
میشه Halt را کاهش می‌دهیم یعنی روشی که در روشی D معادل TM توصیف شده زبان L می‌اندازیم فرض می‌کنیم D یک decidable باشد که ورودی $\langle M, w \rangle$ را می‌گیرد و اگر M روی w می‌تواند طول x متوقف شد، $\langle M, w \rangle$ را می‌گیرد

حال می‌دانی Halt را که می‌دانیم undecidable است. در می‌دانی halt اگر M با ورودی w halt کند accept می‌کنیم و اگر M halt نکند reject می‌کنیم با $1 \leq p$ می‌دانیم که D نمی‌تواند D را decidable مد نظر بگیریم این موضوع نقض می‌شود



امیدوارم که یک مدول مثل M برای D بسازیم که M روی مدول K متوقف شود اگر و تنها اگر M روی w halt کند (متوقف شود)

در واقع متوقف M را برای ورودی x به گذراندن می‌دهیم



با تعریف M مطابق فوق، یک D (بنوان D decider) استی $HALT$ را $decide$ کنیم
 که این یک تناقض است پس D یک $decider$ نیست!

(۷) امتحانی

دو زبان تشخیص پذیر مثل L_1 و L_2 را در نظر می گیریم که هر دو نیز recognizable هستند
حال T_1 و T_2 را ماشین های تشخیص دهنده برای L_1 و L_2 در نظر می گیریم.
زبان L را زبان حاصل از concat کردن L_1 و L_2 در نظر می گیریم
 $L = L_1 \cup L_2$

ماشین جدید T را برای L در نظر می گیریم.
توصیف T :

برای ورودی w در T داریم،
 w را به دو بخش x و y می شکستیم. (w, x, y) مسیری روی حروف نشانی در w را
میزانی T_1 و T_2 را اجرا می کنیم. در واقع تمامی حالت های ممکن برای شکستن w به
 x و y را در نظر گرفته و ماشین های T_1 و T_2 را به طور موازی روی این حالت ها اجرا می کنیم
(حالت ها را به صورت $non-deterministic$ انتخاب می کنیم) چون $non-deterministic$ است
اگر یکی از مسیری ها نیز x و y را بپذیرند، w را می پذیریم (هم T_1 و T_2 را بپذیرد و هم T_1 یا T_2 را)
در صورتی که تمامی سلسله های w در T_1 و T_2 $reject$ شوند ورودی w را
 $reject$ می کنیم. همچنین چون باید برای $reject$ کردن تمامی سیرهای w (تایید هر یکی از آنها)
 $reject$ کنند (ما برای $accept$ ، $accept$ شدن برای یک حالت نیز کافی است پس اگر هیچ
یکی $accept$ نشود و برخی از حالت ها $reject$ شوند در T_1 یا T_2 شوند پس مشخص نبودن
ترتیب T نیز در p ها می ماند. پس T $accept$ یا $reject$ یا p ها نمی در واقع
 $halt$ یا p ها پس T یک ماشین recognizer و L یک زبان تشخیص پذیر است
recognizable است!